Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1906	transparent near3 (host or server)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 11:37
L2	507	virtual adj2 volume$2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 11:38
L3	2014	physical adj2 volume$2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:07
L4	53	l2 same l3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 11:38
L6	3	l4 and l1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:11
L7	118674	replicat$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 11:40
L8	16	l2 same l7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 11:40
L9	21	(fault or failure or fail$3 or bad or defect$) near2 l3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:14
L10	1	l1 and l9	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 11:42

L11	19607	physical adj2 (volume$2 or storage$2 or memor$2 or media$2)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:08
L12	17028	(virtual or logical) adj2 (volume$2 or storage$2 or memor$2 or media$2)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:09
L13	3037	l11 same l12	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:09
L14	569539	translat$4 or map$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:10
L15	1646	l14 same l13	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:10
L16	414669	emulat$4 or simulat$4 or imitat$4 or replicat$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:14
L17	586	l15 and l16	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:14
L18	32	l17 and l1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:42
L19	214	(fault or failure or fail$3 or bad or defect$) near2 l11	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:23
L20	3	l18 and l19	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:15

L21	1345520	fault or failure or fail$3 or bad or defect$	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:23
L22	3	l18 and l20	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:23
L23	303	((first or original) adj (virtual or logical) adj2 (volume$2 or storage$2 or memor$3 or media))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:44
L24	275	(second$3 adj (virtual or logical) adj2 (volume$2 or storage$2 or memor$3 or media))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:45
L25	154	l23 same l24	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:46
L26	39	l25 same l14	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 14:02
L27	4	l26 and l1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:46
L28	108	l25 and l14	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 14:00
L29	0	l25 same l1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 14:02
L30	16	l25 and l1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 14:02

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1906	transparent near3 (host or server)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 11:37
L2	507	virtual adj2 volume$2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 11:38
L3	2014	physical adj2 volume$2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:07
L4	53	l2 same l3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 11:38
L6	3	l4 and l1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:11
L7	118674	replicat$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 11:40
L8	16	l2 same l7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 11:40
L9	21	(fault or failure or fail$3 or bad or defect$) near2 l3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:14
L10	1	l1 and l9	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 11:42

L11	19607	physical adj2 (volume$2 or storage$2 or memor$2 or media$2)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:08
L12	17028	(virtual or logical) adj2 (volume$2 or storage$2 or memor$2 or media$2)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:09
L13	3037	l11 same l12	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:09
L14	569539	translat$4 or map$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:10
L15	1646	l14 same l13	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:10
L16	414669	emulat$4 or simulat$4 or imitat$4 or replicat$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:14
L17	586	l15 and l16	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:14
L18	32	l17 and l1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:42
L19	214	(fault or failure or fail$3 or bad or defect$) near2 l11	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:23
L20	3	l18 and l19	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:15

L21	1345520	fault or failure or fail$3 or bad or defect$	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:23
L22	3	l18 and l20	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:23
L23	303	((first or original) adj (virtual or logical) adj2 (volume$2 or storage$2 or memor$3 or media))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:44
L24	275	(second$3 adj (virtual or logical) adj2 (volume$2 or storage$2 or memor$3 or media))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:45
L25	154	l23 same l24	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:46
L26	39	l25 same l14	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:46
L27	4	l26 and l1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/05 12:46

US-PAT-NO: 5963971

DOCUMENT-IDENTIFIER: US 5963971 A

TITLE: Method and apparatus for handling audit requests of logical volumes in a virtual media server

DATE-ISSUED: October 5, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Fosler; Christine Lynette	Tucson	AZ	N/A	N/A
Gallo; Frank David	Tucson	AZ	N/A	N/A
James; Raymond Anthony	Tucson	AZ	N/A	N/A
Lambert; Anthony Andrew	Tucson	AZ	N/A	N/A

US-CL-CURRENT: 711/114, 711/111

ABSTRACT:

In a data storage subsystem a virtual removable media server (VRMS) handles host audit requests, whether directed to a **physical volume or a virtual volume** physically stored in cache or in a removable physical media item. The subsystem includes a storage interface between the host and data storage facilities including a cache and a physical media library. To the host, the storage interface emulates a virtual library including a plurality of virtual media items, each containing a logical volume of data. The library may also maintain physical media items each storing one logical volume, as in a traditional library. To verify presence of a specified logical volume of the virtual library, the host sends an audit request. Responsively, the subsystem cross-references the specified logical volume in a first list to identify physical locations of underlying data. This physical location may be an address in the cache and/or a **physical volume** corresponding to a physical media item in the library. If the underlying data is stored exclusively in cache, the subsystem automatically answers affirmatively to the audit request. If the underlying data is stored in the physical media, the subsystem searches the library for the physical media item containing the **physical volume** that includes the specified logical volume, and answers the audit request affirmatively if the physical media item is found. If the physical media item is not found, the subsystem answers the audit request negatively.

48 Claims, 6 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 5

---------- KWIC ---------

Abstract Text - ABTX (1):

In a data storage subsystem a virtual removable media server (VRMS) handles host audit requests, whether directed to a **physical volume or a virtual volume** physically stored in cache or in a removable physical media item. The subsystem includes a storage interface between the host and data storage facilities including a cache and a physical media library. To the host, the storage interface emulates a virtual library including a plurality of virtual

media items, each containing a logical volume of data. The library may also maintain physical media items each storing one logical volume, as in a traditional library. To verify presence of a specified logical volume of the virtual library, the host sends an audit request. Responsively, the subsystem cross-references the specified logical volume in a first list to identify physical locations of underlying data. This physical location may be an address in the cache and/or a **physical volume** corresponding to a physical media item in the library. If the underlying data is stored exclusively in cache, the subsystem automatically answers affirmatively to the audit request. If the underlying data is stored in the physical media, the subsystem searches the library for the physical media item containing the **physical volume** that includes the specified logical volume, and answers the audit request affirmatively if the physical media item is found. If the physical media item is not found, the subsystem answers the audit request negatively.

Brief Summary Text - BSTX (3):

The present invention relates to digital data storage systems. More particularly, the invention concerns a data storage subsystem comprising a virtual removable media server (VRMS) that handles audit requests, whether directed to a **physical volume or a virtual volume** physically stored in cache or in a removable physical media item.

Brief Summary Text - BSTX (13):

A data storage subsystem includes a virtual removable media server (VRMS) that handles audit requests from a host, whether the audit requests are directed to a **physical volume or a virtual volume** physically stored in cache or in a removable physical media item. The data storage subsystem includes a storage interface between the host and various data storage facilities including a direct-access storage device (DASD) cache and a physical media library. In communicating with the host, the storage interface emulates a virtual library including a plurality of virtual media items, each containing a logical volume of data. These logical volumes are actually stored in the cache and the library's physical media items. In the library each physical media items may contain multiple logical volumes stacked together, or alternatively, a single logical volume as in a traditional library.

Detailed Description Text - DETX (43):

In addition to the various hardware embodiments described above, a different aspect of the invention concerns a method for handling audit requests in a VRMS, whether such requests are directed to a **physical volume or a virtual volume** physically stored in cache or in a removable physical media item.

Detailed Description Text - DETX (49):

After the routine 400 begins in step 402, the storage interface 104 receives an audit request from the host 102 in step 404. The audit request identifies a particular logical volume whose presence in the subsystem 100 is to be confirmed in response to the audit request. As an example, the identified logical volume may be contained in the cache 106 and/or stacked on a physical media item along with other **virtual volumes**, according to the VRMS aspect of the subsystem 100. The identified logical volume may also correspond to a single **physical volume** in the library 108, maintained by the library 108 as in a traditional library. In the illustrated example, the host 102 identifies the logical volume to be audited by a logical volser, uniquely identifying the logical volume.

Detailed Description Text - DETX (57):

According to the VRMS aspect of the subsystem 100, the identified volume may be a logical volume, contained in the cache 106 or stored alone or stacked on a physical media item along with other **virtual volumes** as space permits. Instead, the identified volume may directly identify a **physical volume** of the library 108, apart from the VRMS aspect of the invention. Thus, the host 102 treats all audit requests alike, the means of physical storage of logical and **physical volumes** being completely **transparent to the host** 102.

US-PAT-NO: 6216202

DOCUMENT-IDENTIFIER: US 6216202 B1

TITLE: Method and apparatus for managing virtual storage devices in a storage system

DATE-ISSUED: April 10, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
D'Errico; Matthew J.	Southboro	MA	N/A	N/A

US-CL-CURRENT: 711/112, 711/114

ABSTRACT:

A method and apparatus for managing a plurality of logical volumes in a computer system, the computer system including a processor and a storage system coupled to the processor, the storage system including at least one storage device, the storage system storing the plurality of logical volumes on the at least one storage device. At least two of the plurality of logical volumes are combined in the storage system into a virtual volume that is presented to the processor as a single logical volume. The storage system also presents the processor with information that enables the processor to deconstruct the virtual volume into the at least two of the plurality of logical volumes. Another aspect is directed to a multi-path computer system including a processor, a storage system including at least one storage device to store Y logical volumes, and X paths coupling the processor to the storage system. The processor is capable of accessing each of the Y logical volumes through each of the X paths, and includes Z unique target address identifiers identifying the Y logical volumes, wherein Z is less than X times Y.

53 Claims, 5 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 5

---------- KWIC ---------

Brief Summary Text - BSTX (4):

Many computer systems include one or more host computers and one or more storage systems that store data used by the host computers. An example of such a system is shown in FIG. 1, and includes a host computer 1 and a storage system 3. The storage system typically includes a plurality of storage devices on which data is stored. In the exemplary system shown in FIG. 1, the storage system includes a plurality of disk drives 5a-b, and a plurality of disk controllers 7a-7b that respectively control access to the disk drives 5a and 5b. The storage system 3 further includes a plurality of storage bus directors 9 that control communication with the host computer 1 over communication buses 17. The storage system 3 further includes a cache 11 to provide improved storage system performance. In particular, when the host computer 1 executes a read from the storage system 3, the storage system 3 may service the read from the cache 11 (when the data is stored in the cache), rather than from one of the disk drives 5a-5b, to execute the read more efficiently. Similarly, when

the host computer 1 executes a write to the storage system 3, the corresponding storage bus director 9 can execute the write to the cache 11. Thereafter, the write can be destaged asynchronously, in a manner **transparent to the host** computer 1, to the appropriate one of the disk drives 5a-5b. Finally, the storage system 3 includes an internal bus 13 over which the storage bus directors 9, disk controllers 7a-7b and the cache 11 communicate.

Claims Text - CLTX (12):
the step (A) includes a step of combining, in the storage system, a first pair of the plurality of logical volumes into a **first virtual volume** that includes the first pair of the plurality of logical volumes, and a step of combining, in the storage system, a second pair of the plurality of logical volumes into a **second virtual volume** that includes the second pair of the plurality of logical volumes;

Claims Text - CLTX (38):
means for combining a first pair of the plurality of logical volumes into a **first virtual volume** that includes the first pair of the plurality of logical volumes, and for combining a second pair of the plurality of logical volumes into a **second virtual volume** that includes the second pair of the plurality of logical volumes;

US-PAT-NO: 6643667

DOCUMENT-IDENTIFIER: US 6643667 B1

TITLE: System and method for replicating data

DATE-ISSUED: November 4, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Arai; Kouji	Odawara	N/A	N/A	JP
Suzuki; Susumu	Oiso	N/A	N/A	JP
Yasukawa; Hironori	Odawara	N/A	N/A	JP

APPL-NO: 09/ 528416

DATE FILED: March 17, 2000

FOREIGN-APPL-PRIORITY-DATA:

COUNTRY	APPL-NO	APPL-DATE
JP	11-075174	March 19, 1999

US-CL-CURRENT: 707/200, 707/1 , 710/33 , 711/112 , 711/114 , 711/167 , 714/6

ABSTRACT:

According to the present invention, techniques for controlling copying of logical volumes within a computer storage system are provided. A representative embodiment includes a plurality of storage devices controlled by a control unit, one or more processors, and a buffer memory for temporarily storing data read from the storage devices within the control unit. The storage devices can be addressed as logical volumes.

37 Claims, 18 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 15

---------- KWIC ---------

Brief Summary Text - BSTX (10):

In an exemplary embodiment, the invention provides a method for creating a copy on a **second logical volume** of data stored on a **first logical volume**. The method can comprise a variety of steps, such as specifying a relationship between two or more logical volumes. The method can also include creating a copy of data in a specified **first logical volume into said second logical volume**. Creating such a copy can include steps of copying data from the **first logical volume** to a first location in a buffer memory located within a control unit. Copying can be performed by the control unit substantially independently of processor control. Then, data can be copied from the first location in the buffer memory to a second location in the buffer memory. Subsequently, data from the second location in the buffer memory can be copied to the **second logical volume**. This copying can be performed by the control unit substantially independently of processor control, also. As used herein, substantially independently of processor control can include performing copy

processing at the control unit level without necessitating intermediate communication between a command start from the processor to the control unit and a command complete signal from the control unit to the processor.

Brief Summary Text - BSTX (11):

In another embodiment, the invention provides a computer system comprising a plurality of devices. A plurality of storage devices controlled by one or more control units can be part of the computer system. One or more processing units operable to access the control unit or units can also exist in the computer system. A buffer memory operable to temporarily store data read from the storage devices within the control unit can also be part of the computing system. The storage devices can be addressed as one or more logical volumes. The control unit is able to establish a relationship between at least two logical volumes (i.e., a **first logical volume and a second logical volume**) located in the storage devices. The control unit can create a copy of data in the **first logical volume into the second logical volume**. Such creating a copy can include copying data from the **first logical volume** to a first location in the buffer memory. Then, the data can be copied from the first location in the buffer memory to a second location in the buffer memory, changing meta-data indicating the device that may access the data to reflect the **second logical volume**. Thereupon, the data can be copied from the second location in the buffer memory to the **second logical volume**. These operations by the control unit can be performed substantially independently of the processing units. In a representative embodiment, the buffer can comprise approximately 10 Gigabytes, for example.

Brief Summary Text - BSTX (12):

In a further embodiment, the invention provides a computer program product for controlling the copying of information from a **first logical volume to a second logical volume** in a computer system. The computer program product can comprise a computer readable storage medium containing a variety of program code. Code for specifying a relationship between the **first logical volume and the second logical volume** can be part of the computer program product. The product can also include code for creating a copy of data in the **first logical volume into the second logical volume**. The code for creating a copy can comprise various program codes. Program code for copying data from the **first logical volume** to a first location in a buffer memory can be part of the program product. The product can also include code for copying the data from the first location in the buffer memory to a second location in the buffer memory. Code for copying the data from the second location in the buffer memory to the **second logical volume** can also part of the program product. The codes for copying the data from the first location in the buffer memory to the second location in the buffer memory is executed by a control unit substantially independently of a central processing unit.

Detailed Description Text - DETX (8):

Re-synchronize pair command 214 can bring a status transition to "paired and not copied" status again after a copy is created. In a presently preferred embodiment, re-synchronize pair processing can compare the secondary volume track **map** with the primary volume track **map** in control information 401 of FIG. 4 in order to determine all unequal tracks. Then, unequal tracks can be copied from the primary volume to the secondary volume. As with create pair command 211 processing, a pace may be selected for the re-synchronize copy operations from among slow, medium and fast.

Claims Text - CLTX (1):

1. A method for creating a copy of data in a system comprising a plurality of storage devices, a control unit operable to control said storage devices, and a memory operable to temporarily store data read from said storage devices within said control unit, said storage devices addressable as at least one of a plurality of logical volumes, including a **first logical volume and a second logical volume**, said method comprising: specifying a relationship between at least two of said logical volumes, said relationship defined between said **first logical volume and said second logical volume**; creating a copy of data in said specified **first logical volume into said second logical volume**; said creating a copy further comprising: copying data from said **first logical volume** to a first location in said memory; copying said data from said first location in said memory to a second location in said memory; copying said data from said second location in said memory to said **second logical volume**; wherein said copying said data from said first location in said memory to a second location in said memory is performed by said control unit.

Claims Text - CLTX (2):
2. The method of claim 1, wherein said copying said data from said first location in said memory to a second location in said memory further comprises: reading data from said first location in said memory into a location within an address change unit; exchanging a logical address within said data from an address corresponding to said **first logical volume** to an address corresponding to said **second logical volume**; and writing said data to said second location in said memory.

Claims Text - CLTX (3):
3. The method of claim 1 further comprising: if a write request is issued to said **first logical volume** after creating a copy has commenced, creating a copy of data in said **first logical volume to said second logical volume** before said data in said **first logical volume** is modified by said write request.

Claims Text - CLTX (6):
6. The method of claim 1 further comprising: making said **second logical volume** accessible after said creating a copy of data in said specified **first logical volume into said second logical volume**.

Claims Text - CLTX (7):
7. The method of claim 1 further comprising: tracking modified data, if a write request is issued to said **first logical volume or said second logical volume** after the copy processing is completed, and copying said modified data based upon said tracking, if creating a copy is directed again to the pair in copy completed status.

Claims Text - CLTX (9):
9. The method of claim 1 wherein said **first logical volume** is defined as a primary logical volume, said method further comprising: defining at least one of a plurality of different logical volumes as **secondary logical volumes**; and defining multiple pairs comprising said primary logical volume and one of said plurality of **secondary logical volumes**.

Claims Text - CLTX (11):
11. The method of claim 1 further comprising: displaying information about said **first logical volume and said second logical volume**.

Claims Text - CLTX (12):
12. A method for controlling the copying of information from a **first logical volume to a second logical volume** in a computer system, said method comprising: specifying a relationship between said **first logical volume and said second logical volume**; creating a copy of data in said **first logical volume into said second logical volume**; said creating a copy further comprising: copying data from said **first logical volume** to a first location into a memory; copying said data from said first location in said memory to a second location in said memory; copying said data from said second location in said memory to said **second logical volume**; wherein said copying said data from said first location in said memory to a second location in said memory is performed by a control unit substantially independently of a central processing unit.

Claims Text - CLTX (13):
13. A method for controlling the copying of information from a **first logical volume to a second logical volume** in a computer system, said method comprising: specifying a relationship between said **first logical volume and said second logical volume**; copying data read from said **first logical volume** into a memory located within a control unit and thereupon writing said data to said **second logical volume**; and wherein said copying said data from a first location in said memory to a second location in said memory is performed by a control unit substantially independently of a central processing unit.

Claims Text - CLTX (14):
14. A computer system comprising a plurality of storage devices, a control unit operable to control said storage devices, and a memory operable to temporarily store data read from said storage devices within said control unit, said storage devices addressable as at least one of a plurality of logical volumes, including a **first logical volume and a second logical volume,** said control unit operatively disposed to: establish a relationship between at least two of said logical volumes, said relationship defined between said **first logical volume and said second logical volume**; create a copy of data in said specified **first logical volume into said second logical volume**; said creating a copy further comprising: copy data from said **first logical volume** to a first location in said memory; copy said data from said first location in said memory to a second location in said memory; copy said data from said second location in said memory to said **second logical volume**; wherein said copy said data from said first location in said memory to a second location in said memory is performed by said control unit.

Claims Text - CLTX (15):
15. The computing system of claim 14 wherein said copy said data from said first location in said memory to a second location in said memory further comprises: reading data from said first location in said memory into a location within an address change unit; exchanging a logical address within said data from an address corresponding to said **first logical volume** to an address corresponding to said **second logical volume**; and writing said data to said second location in said memory.

Claims Text - CLTX (20):
20. A computer program product for controlling the copying of information from a **first logical volume to a second logical volume** in a computer system, said computer program product comprising: code for specifying a relationship between said **first logical volume and said second logical volume**; code for

creating a copy of data in said **first logical volume into said second logical volume**; said code for creating a copy further comprising: code for copying data from said **first logical volume** to a first location into a memory; code for copying said data from said first location in said memory to a second location in said memory; code for copying said data from said second location in said memory to said **second logical volume**; wherein said copying said data from said first location in said memory to a second location in said memory is performed by a control unit substantially independently of a central processing unit; and a computer readable storage medium for holding the codes.

Claims Text - CLTX (21):
21. A computer program product for controlling the copying of information from a **first logical volume to a second logical volume** in a computer system, said computer program product comprising: code for specifying a relationship between said **first logical volume and said second logical volume**; code for copying data read from said **first logical volume** into a memory located within a control unit and thereupon writing said data to said **second logical volume**; and wherein said copying said data from said first location in said memory to a second location in said memory is performed by said control unit substantially independently of a central processing unit; and a computer readable storage medium for holding the codes.

Claims Text - CLTX (22):
22. The computer program product of claim 21 further comprising: code for displaying information about said **first logical volume and said second logical volume**.

Claims Text - CLTX (23):
23. A control unit for controlling the copying of information, said control unit operable in a computing system comprising at least one of a plurality of storage devices, said control unit operable to control said storage devices, at least one of a plurality of processing units operable to access said control unit, said storage devices addressable as at least one of a plurality of logical volumes, including a **first logical volume and a second logical volume,** said control unit comprising a memory operable to temporarily store data read from said storage devices within said control unit, said control unit operatively disposed to: copy data read from said **first logical volume** into said memory located within said control unit; copy said data from said memory to a different location within said memory, changing a volume identifier associated with said data, and thereupon writing said data to said **second logical volume**; and wherein said copying said data from a first location in said memory to a second location in said memory is performed by said control unit substantially independently of a central processing unit.

Claims Text - CLTX (24):
24. A computer system comprising a plurality of storage devices, said storage devices addressable as at least one of a plurality of logical volumes, including a **first logical volume and a second logical volume,** at least one of a plurality of processing units, a cache memory operable to temporarily store data, and a control unit operable to store and retrieve data from said storage devices on behalf of said processing units; wherein said control unit is further operable to copy data from a **first logical volume to a second logical volume** according to a relationship established between said **first logical volume and said second logical volume**; wherein said control unit copies said data from said **first logical volume** to a first location in said cache memory; whereupon a data recovery unit within said control unit is operable to create a

copy of said data in said first location in said cache memory to a buffer memory within said data recovery unit, and thereupon to copy said data from said buffer memory within said data recovery unit into a second location in said cache memory; and thereupon to copy said data from said second location in said cache memory to said **second logical volume**; wherein said data comprises a logical address section, said logical address section having a data content that is changed during said copying between said cache memory and said memory.

Claims Text - CLTX (27):
27. The method of claim 1, wherein said control unit comprises at least one disk adapter, and wherein said at least one disk adapter performs the step of creating a copy of data in said specified **first logical volume into said second logical volume**.

Claims Text - CLTX (30):
30. The method of claim 14, wherein said control unit comprises at least one disk adapter, and wherein said at least one disk adapter is configured to create said copy of data in said specified **first logical volume into said second logical volume**.

Claims Text - CLTX (32):
32. A method for creating a copy of data in a system comprising a plurality of storage devices, a control unit operable to control said storage devices, said control unit comprising at least one disk adapter and a memory operable to temporarily store data read from said storage devices within said control unit, said storage devices addressable as at least one of a plurality of logical volumes, including a **first logical volume and a second logical volume,** said method comprising: specifying a relationship between at least two of said logical volumes, said relationship defined between said **first logical volume and said second logical volume**; said at least one disk adapter creating a copy of data in said specified **first logical volume into said second logical volume**; said creating a copy further comprising: copying data from said **first logical volume** to a first location in said memory; copying said data from said first location in said memory to a second location in said memory; copying said data from said second location in said memory to said **second logical volume**.

Claims Text - CLTX (34):
34. The method of claim 32, wherein said disk adapter comprises an address change unit, and wherein said copying said data from said first location in said memory to a second location in said memory further comprises: reading data from said first location in said memory into a location within said address change unit; exchanging a logical address within said data from an address corresponding to said **first logical volume** to an address corresponding to said **second logical volume**; and writing said data to said second location in said memory.

Claims Text - CLTX (35):
35. A computer system comprising a plurality of storage devices, a control unit operable to control said storage devices, said control unit comprising at least one disk adapter and a memory operable to temporarily store data read from said storage devices within said control unit, said storage devices addressable as at least one of a plurality of logical volumes, including a **first logical volume and a second logical volume,** said at least one disk adapter operatively disposed to: establish a relationship between at least two

of said logical volumes, said relationship defined between said **first logical volume and said second logical volume**; create a copy of data in said specified **first logical volume into said second logical volume**; said creating a copy further comprising: copy data from said **first logical volume** to a first location in said memory; copy said data from said first location in said memory to a second location in said memory; copy said data from said second location in said memory to said **second logical volume**.

Claims Text - CLTX (37):

37. The method of claim 35, wherein said disk adapter comprises an address change unit, and wherein said copying said data from said first location in said memory to a second location in said memory further comprises: reading data from said first location in said memory into a buffer location within said address change unit; exchanging a logical address within said data from an address corresponding to said **first logical volume** to an address corresponding to said **second logical volume**; and writing said data to said second location in said memory.

US-PAT-NO: 6606690

DOCUMENT-IDENTIFIER: US 6606690 B2

TITLE: System and method for accessing a storage area network as network attached storage

DATE-ISSUED: August 12, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Padovano; Michael	Englishtown	NJ	N/A	N/A

US-CL-CURRENT: 711/148, 709/213 , 709/218 , 709/220 , 709/249 , 711/170

ABSTRACT:

A method, system, and apparatus for accessing a plurality of storage devices in a storage area network (SAN) as network attached storage (NAS) in a data communication network is described. A SAN server includes a first interface and a second interface. The first interface is configured to be coupled to the SAN. The second interface is coupled to a first data communication network. A NAS server includes a third interface and a fourth interface. The third interface is configured to be coupled to a second data communication network. The fourth interface is coupled to the first data communication network. The SAN server allocates a first portion of the plurality of storage devices in the SAN to be accessible through the second interface to at least one first host coupled to the first data communication network. The SAN server allocates a second portion of the plurality of storage devices in the SAN to the NAS server. The NAS server configures access to the second portion of the plurality of storage devices to at least one second host coupled to the second data communication network.

37 Claims, 32 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 29

---------- KWIC ---------

Detailed Description Text - DETX (36):

A storage appliance, such as the SANLink.TM., unifies SAN management by providing resource allocation to hosts. In the case of the SANLink.TM., for instance, it also provides data management capabilities. These data management capabilities may include: 1. Storage virtualization/**mapping**. All connected storage in the SAN is provided as a single pool of storage, which may be partitioned and shared among hosts as needed. 2. Data mirroring. An exact copy of the data stored in the SAN storage devices is created and maintained in real time. The copy may be kept at a remote location. 3. Point-in-time copying (Snapshot). An instantaneous virtual image of the existing storage may be created. The virtual replica can be viewed and manipulated in the same way as the original data. 4. Storage security. Access to particular storage devices, or portions thereof, may be restricted. The storage devices or portions may be masked from view of particular hosts or users.

Detailed Description Text - DETX (85):

SAN storage manager 404 provides data management functionality for SAN server 302. SAN storage manager 404 includes one or more modules that are directed towards controlling aspects of data management for the SAN. In the embodiment shown in FIG. 4, SAN storage manager 404 includes a storage allocator module 416, a storage **mapper** module 418, a data mirror module 420, a snapshot module 422, and a storage security module 424.

Detailed Description Text - DETX (87):

Storage **mapper** module 418 controls the **mapping** of logical storage addresses received from hosts 102, 104, and 106, and from NAS server 304, to actual physical storage addresses for data stored in the storage devices of SAN 120.

Detailed Description Text - DETX (92):

SAN storage manager 404 receives the read and write storage requests from second network interface 402, and processes them accordingly. For instance, SAN storage manager 404 may **map** the received storage request from a logical storage address to one or more physical storage address. The SAN storage manager 404 outputs the physical storage address(s) to first network interface 406.

Detailed Description Text - DETX (112):

In an embodiment, an existing administrative interface that accommodates SAN servers may not require any modification to handle allocation of storage to NAS servers. However, although the SAN servers view the NAS servers as separate hosts, in an embodiment, an existing administrative interface may be enhanced to allow integration of NAS functionality. For example, in an embodiment, the administrative interface may be configured to show the NAS servers as themselves, rather than as hosts. For instance, the administrative interface may allow the storage appliance administrator to allocate a storage portion, such as a LUN, to a NAS server, as a NAS LUN. Any LUN **mapping** may be done automatically. Once the administrator chooses a LUN to be a NAS LUN, the NAS servers create a file system for that LUN and export that file system to the network. This process is described in further detail below.

Detailed Description Text - DETX (144):

In step 1602, the NAS servers are found within one or more host **mapping** tables. SAN server 302a does this by looking up the special names of the NAS servers registered with SAN server 302a. The names of the NAS servers are registered with SAN servers 302a and 302b when the NAS servers boot up, as described above. For example, the registered names of NAS servers 304a and 304b are NASServerNASOne and NASServerNASTwo.

Detailed Description Text - DETX (145):

In step 1604, the value of the second parameter is determined, indicating whether the LUN is enabled. If the second parameter is a 1, SAN server 302a **maps** the LUN to NASServerNASOne and NASServerNASTwo. The LUN is **mapped** to both servers in the event of fail-over, as described in further detail below. If SAN server 302a determines that the second parameter is a 0, indicating that the LUN is not enabled, SAN server 302a removes the LUN from the host **map** for NAS servers 304a and 304b.

Detailed Description Text - DETX (153):

That instructs SAN server 302a to allocate LUN 15 to NAS server 304b. After **mapping** the LUN to both NAS servers (as described in steps 1 through 4 above), SAN server 302a sends the following string to NAS server 304a: LUN:Enable:15:0

Detailed Description Text - DETX (165):

In an embodiment, a NAS server uses a Linux operating system. The Linux operating system uses symbolic names for disk devices, such as /dev/sda, /dev/sdb, and /dev/sdc. Because of this, it is difficult to determine the LUN number from the symbolic name. To overcome this difficulty, the NAS server maintains a **map** of LUN numbers to symbolic names. For example, the **map** may be maintained via Linux symbolic links. The symbolic links may be kept in a directory named /dev/StorageDir, and contain the HBA number, the controller number, the target number, and the LUN number.

Detailed Description Text - DETX (166):

For example, NAS server 304a may receive a directive to enable LUN 15. After NAS server 304a configured LUN 15 on HBA 1 into the kernel of operating system 510 of NAS server 304a, the kernel may assign LUN 15 the device name of /dev/sde. To maintain the **mapping** of LUN 15 on HBA 1 to device /dev/sde, NAS server 304a may create a symbolic link to /dev/sde named /dev/StorageDir/1.0.0.15. Subsequent NAS operations may be performed on /dev/StorageDir/1.0.0.15.

Detailed Description Text - DETX (315):

In step 2512, a Gratuitous ARP request is issued, which allows clients to update their IP-to-Ethenet **mapping** information.

Detailed Description Text - DETX (349):

In step 2804, a Gratuitous ARP request is issued, causing clients/hosts to update their IP-to-Ethernet **mapping** tables.

Claims Text - CLTX (21):

21. The apparatus of claim 20, wherein said first portion of the plurality of storage devices in the SAN includes a first at least one physical storage device, wherein said SAN server further includes: a storage **mapper that maps** said first at least one physical storage device to at least one **first logical storage** device that is accessible to said at least one first host; wherein said second portion of the plurality of storage devices in the SAN includes a second at least one physical storage device; and wherein said storage **mapper maps** said second at least one physical storage device to at least one **second logical storage** device that is allocated to said NAS server.

US-PAT-NO: 6606690

DOCUMENT-IDENTIFIER: US 6606690 B2

TITLE: System and method for accessing a storage area network as network attached storage

DATE-ISSUED: August 12, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Padovano; Michael	Englishtown	NJ	N/A	N/A

US-CL-CURRENT: 711/148, 709/213 , 709/218 , 709/220 , 709/249 , 711/170

ABSTRACT:

A method, system, and apparatus for accessing a plurality of storage devices in a storage area network (SAN) as network attached storage (NAS) in a data communication network is described. A SAN server includes a first interface and a second interface. The first interface is configured to be coupled to the SAN. The second interface is coupled to a first data communication network. A NAS server includes a third interface and a fourth interface. The third interface is configured to be coupled to a second data communication network. The fourth interface is coupled to the first data communication network. The SAN server allocates a first portion of the plurality of storage devices in the SAN to be accessible through the second interface to at least one first host coupled to the first data communication network. The SAN server allocates a second portion of the plurality of storage devices in the SAN to the NAS server. The NAS server configures access to the second portion of the plurality of storage devices to at least one second host coupled to the second data communication network.

37 Claims, 32 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 29

---------- KWIC ---------

Claims Text - CLTX (21):

21. The apparatus of claim 20, wherein said first portion of the plurality of storage devices in the SAN includes a first at least one physical storage device, wherein said SAN server further includes: a storage **mapper that maps** said first at least one physical storage device to at least one **first logical storage** device that is accessible to said at least one first host; wherein said second portion of the plurality of storage devices in the SAN includes a second at least one physical storage device; and wherein said storage **mapper maps** said second at least one physical storage device to at least one **second logical storage** device that is allocated to said NAS server.